EXHIBIT A

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

SAN FRANCISCO DIVISION

ORACLE AMERICA, INC. Plaintiff, v. GOOGLE INC. Defendant.	Case No. CV 10-03561 WHA

OPENING EXPERT REPORT OF ALAN PURDY REGARDING COPYRIGHT

SUBMITTED ON BEHALF OF PLAINTIFF ORACLE AMERICA, INC.

Comparison of Oracle's Java Library API Specifications Versus Android's Library API Specifications

Alan Purdy
July 29, 2011

P.O. Box 10666
Portland OR 97296
Tel: (503) 274-0784
Cell: (503) 784-5907

alan@jli.com

Introduction

1. My name is Alan Purdy. I am a Senior Forensic Software Analyst at Johnson-Laird, Inc. ("JLI").

2. I have been retained as a consultant in this action by counsel for Oracle Corporation.

3. JLI is an Oregon-based corporation that provides consulting services to computer hardware and software manufacturers, and computer-related technical assistance to the legal profession in the United States, Canada, Japan, Singapore, and Europe. JLI specializes in providing consulting services to corporations and attorneys on intellectual property matters (such as forensic analysis of computer-related evidence, copyright and patent infringement, misappropriation of trade secrets, and software due diligence for mergers and acquisitions) and performing assessments of computer software and Techno-archeology™ (*i.e.*, the analysis of software development projects). My *curriculum vitae* is attached as Exhibit 1 to this Report.

4. This Report represents my current opinions based upon the information that I have been supplied—cited herein—and certain analysis that I have been able to perform thus far. I reserve the right to supplement this Report with any additional information provided to me, or if I should be asked to perform additional analysis beyond that which is described below.

5. JLI is compensated at the rate of $450 per hour for my work in this matter. This compensation is not conditioned on the outcome of this matter.

Background

6. Oracle provides to Java developers a set of core libraries that include classes of code that perform common functions. Java developers create applications that rely on the Java libraries. The Application Program Interface ("API") specifications for these libraries are documented on Oracle's website. (*See, e.g.*, http://download.oracle.com/javase/1.5.0/docs/api/index.html). Developers read the API specifications to understand the interfaces to the classes and modules included in the libraries.

7. Google supplies a similar website describing Android's class libraries for developing programs for Google's Android platform. (http://developer.android.com/reference/packages.html). Google Android's API specifications currently describe 13 distinct "API Levels" that correspond to different versions of Android.

8. I was retained to analyze similarities between the Android API specifications and the Java API specifications. Specifically, my goal was to tabulate, for certain packages within the Java Standard Edition version 5 Development Kit ("JDK 5"), the number of classes, interfaces, and exceptions that are similar or the same in both platforms. I am informed that Oracle is asserting copyright infringement of the 37 Java packages that I was asked to analyze (the "Analyzed Packages").[1] Additionally, for each class in the Analyzed Packages, I was asked to determine the number of fields, constructors, and method signatures that were the same in both.

9. I was asked to perform comparisons between the following versions of the Java Standard Edition and Android API specifications:

JDK 5 vs. Android API Level 3 ("Cupcake")

JDK 5 vs. Android API Level 8 ("Froyo")

I understand that for developers to build Android applications, Google requires them to download the JDK version 5 or 6. (*See* http://developer.android.com/sdk/requirements.html).

Analysis Methods

10. To perform the requested task, I wrote a Java-based program to extract information from both Oracle's Java JDK API descriptions, and Google's Android SDK library descriptions. More particularly, this program started from the list of Analyzed Packages, reading each corresponding package's web page, and transitively reading all the pages that describe the referenced classes, interfaces, enums, errors, exceptions, and annotation types. Upon visiting each page, the program then captured relevant information for the next steps of the analysis. For example, after reading a page for a specific class, it read and captured the constructors, fields, and methods described on that page. After capturing that information (e.g., such as method signatures), it then wrote the information to a Microsoft Access database for further analysis.

11. After capturing all the necessary information, I then wrote a set of scripts to compare, for example, method, constructor, and field signatures. This document reports on the results of those scripts.

[1] I understand that Oracle is not asserting copyright infringement of certain other packages, in some cases because Oracle either uses them under license from third parties or because Oracle allows third parties to utilize these packages under permissive terms. These packages include: java.math, java.util.concurrent, java.util.concurrent.atomic, java.util.concurrent.locks, javax.xml, javax.xml.datatype, javax.xml.namespace, javax.xml.parsers, javax.xml.transform, javax.xml.transform.dom, javax.xml.transform.sax, javax.xml.transform.stream, javax.xml.validation, and javax.xml.xpath. These packages are not included in my analysis.

Results of Analysis

12. This section presents the results of my analysis in two sections: Analyzed Packages and Analyzed Classes.

Analyzed Packages: Results of Analysis

Packages: Java JDK 5 vs. Android API Level 3

13. My analysis determined the following metrics with respect to Java JDK 5 libraries and Android's API Level 3 packages:

Package Elements Under Analysis	Java 5 Count	Android API Level 3 Count	In Java 5 but not Android 3 Count (%)	In Android 3 but not Java 5 Count	In Android 3 and Java 5 Count (%)
Packages	37	37	0	0	37 (100%)
Annotation Types	7	0	7 (100%)	0	0 (0%)
Classes	506	458	48 (9.4%)	0	458 (90.6%)
Enums	9	9	0 (0%)	0	9 (100%)
Errors	25	25	0 (0%)	0	25 (100%)
Exceptions	176	165	11 (6.25%)	0	165 (93.75%)
Interfaces	171	158	13 (7.6%)	0	158 (92.4%)
Totals	**931**	**852**	**79**	**0**	**852**

Packages: Java JDK 5 vs. Android API Level 8

14. My analysis determined the following metrics with respect to Java JDK 5 libraries and Android's API Level 8 packages:

Package Elements Under Analysis	Java 5 Count	Android API Level 8 Count	In Java 5 but not Android 8 Count (%)	In Android 8 but not Java 5 Count	In Android 8 and Java 5 Count (%)
Packages	37	37	0	0	37 (100%)
Annotation Types	7	0	7 (100%)	0	0 (0%)
Classes	509	458	48 (9.4%)	0	458 (90.6%)
Enums	9	9	0 (0%)	0	9 (100%)
Errors	25	25	0 (0%)	0	25 (100%)
Exceptions	176	165	11 (6.25%)	0	165 (93.75%)
Interfaces	171	158	13 (7.6%)	0	158 (92.4%)
Totals	**934**	**852**	**86**	**0**	**852**

Analyzed Classes: Results of Analysis

Classes: Java JDK 5 vs. Android API Level 3

15. My analysis determined the following metrics with respect to Java JDK 5 libraries and Android's API Level 3 classes. Because, within the Analyzed Packages, Android API Level 3 does not provide 48 of Java 5's classes, the table below only compares the fields, constructors and methods ("Class Elements") of those 458 classes in both libraries. The "Java 5 Count" column below therefore identifies the total number of fields (constructors, methods) that exist in Java 5 classes where that class also exists in Android API Level 3 (for the Analyzed Packages). For the right-most column, I compared the signatures of each Class Element, including the modifiers, names, parameters and types, extends (superclass), and implements (interfaces) between Java 5 and Android 3.

Class Elements Compared	Java 5 Count	Android API Level 3 Count	Java 5 Element Matches Android 3 Count
Fields	1060	927	893
Constructors	642	639	383
Methods	3672	5052	2427
Totals	**5374**	**6618**	**3703**

16. When compared, the numbers of methods in each class for corresponding classes between Java 5 and Android API Level 3 have some differences. The following table has three columns, the left-most identifies a class name in which that Java 5 class has a different number of methods than the same class in Android API Level 3. The middle column shows the number of Android 3's methods in that row's class (*see* the left-most column). The right-most column shows the number of Java 5 methods in that row's class. All other classes (of the 458) shared by both systems have the same number of methods in the corresponding class.

Classes With Different Number of Methods Per Class		
Name	**Android 3 Methods**	**Java 5 Methods**
java.io.BufferedOutputStream	4	3
java.io.RandomAccessFile	40	39
java.lang.Class	53	46
java.lang.Enum	10	9
java.lang.Package	17	18
java.lang.reflect.Array	20	21
java.lang.reflect.Constructor	17	18
java.lang.reflect.Field	30	31

Classes With Different Number of Methods Per Class		
Name	**Android 3 Methods**	**Java 5 Methods**
java.lang.reflect.Method	21	22
java.net.HttpURLConnection	16	17
java.net.Inet4Address	10	14
java.net.Inet6Address	16	19
java.net.Socket	42	11
java.net.SocketPermission	5	1
java.nio.Buffer	14	13
java.nio.ByteBuffer	56	57
java.nio.DoubleBuffer	24	25
java.nio.FloatBuffer	24	25
java.nio.IntBuffer	24	25
java.nio.LongBuffer	24	25
java.nio.ShortBuffer	24	25
java.security.Security	11	7
java.text.CollationKey	3	5
java.text.DateFormat.Field	2	3
java.text.DateFormatSymbols	20	19
java.util.jar.JarFile	6	5
java.util.regex.Pattern	11	10
java.util.TreeMap	16	18
java.util.zip.DeflaterOutputStream	6	5
java.util.zip.ZipInputStream	6	7

Classes: Java JDK 5 vs. Android API Level 8

17. My analysis determined the following metrics with respect to Java JDK 5 libraries and Android's API Level 8 classes. Because Android API Level 8 does not provide 48 of Java 5's classes, the table below only compares the fields, constructors and methods ("Class Elements") of those 461 classes from the Analyzed Packages in both libraries. The "Java 5 Count" column below therefore identifies the total number of fields (constructors, methods) that exist in Java 5 classes where that class also exists in Android API Level 8 (for the Analyzed Packages). For the right-most column, I compared the signatures of each Class Element, including the modifiers, names, parameters and types, extends (superclass), and implements (interfaces) between Java 5 and Android 8.

Class Elements Under Analysis	Java 5 Count	Android API Level 8 Count	Java 5 Element Matches Android 8 Count
Fields	1060	927	893
Constructors	642	639	383
Methods	3672	5052	2427
Total	**5374**	**6618**	**3703**

Submitted on July 29, 2011, in Portland, Oregon

Alan Purdy

Appendix A: Analyzed Packages

java.awt.font

java.beans

java.io

java.lang

java.lang.annotation

java.lang.ref

java.lang.reflect

java.net

java.nio

java.nio.channels

java.nio.channels.spi

java.nio.charset

java.nio.charset.spi

java.security

java.security.acl

java.security.cert

java.security.interfaces

java.security.spec

java.sql

java.text

java.util

java.util.jar

java.util.logging

java.util.prefs

java.util.regex

java.util.zip

javax.crypto

javax.crypto.interfaces

javax.crypto.spec

javax.net

javax.net.ssl

javax.security.auth

javax.security.auth.callback

javax.security.auth.login

javax.security.auth.x500

javax.security.cert

javax.sql

Exhibit 1: Curriculum Vitae & Qualifications

Alan Purdy
July 29, 2011

P.O. Box 10666
Portland OR 97296
Tel: (503) 274-0784
Cell: (503) 784-5907

alan@jli.com

Curriculum Vitae of Alan Purdy

P.O. Box 10666
Portland OR 97296
Tel: (503) 274-0784
Cell: (503) 784-5907
alan@jli.com

Experience Summary

Software forensics analyst – Analyst in cases under non-disclosure concerning patent infringement, patent validity, and trade secrets; analyst in Kodak v. Sun Microsystems patent infringement litigation. Member of and invited speaker to the Information Technology Study Group (ITSG), an FBI-sponsored forum held twice per year.

Technology consultant – Suspended technical consulting in 2000. Prior clients include: Apple, Citicorp, Gemini Consulting, Hughes, Interval Research, IBM Japan, IBM Brazil, Proxima Systems Ltd., Texaco, and USAirways. Specialized in object-oriented systems, database systems, and decision-support systems.

Systems developer – Language compilers, database systems, operating systems, team leader, and project manager.

Computer scientist and software designer – Comshare, Tektronix, GemStone, and Xerox PARC.

Object-oriented systems evangelist – System architect and designer of GemStone's object database management system; chairman of ACM's first OOPSLA conference in 1986. Invited speaker to many object technology forums.

Education

1999-2006 Software forensics training, Johnson-Laird Inc.

1971 B.S. Science Engineering, University of Michigan.

Publications

Purdy, A., Maier, D., and Schuchardt, B. *Integrating an Object Server with Other Worlds.* ACM Transactions on Office Information Systems, 5,1, (Jan. 1987) 27-47.

Maier, D., Stein, J., Otis. A., Purdy, A. *Development of an Object-oriented DBMS.* ACM SIGPLAN Notices 21, 11 (Nov. 1986) 472-482.

Maier, D., Otis, A., and Purdy, A. *Object-oriented Database Development at Servio Logic.* IEEE Database Engineering 8,4 (Dec. 1985) 58-65.

Experience Details

Forensics Analyst	Consulting software forensics analyst on several trade secret, copyright, and software patent cases, under non-disclosure. Analyst in Kodak v. Sun Microsystems patent infringement litigation.
Applications	System architect of a physician's order-entry system in Fukuoka Japan while on contract to IBM-Japan. Developed four business decision support systems for USAirways under contract to their operations research group. Developed several database-driven dynamic web sites.
Database Systems	Member of small team that designed and implemented Comshare's second generation RDBMS. System architect of Gemstone, a multi-user object-oriented database system featuring the Smalltalk language.
Compilers	Compilers, assemblers, debuggers for Pascal, RPG-II, real-time languages, and Smalltalk, for a variety of machines, some of them virtual.
Operating Systems	While at Comshare, co-designed a transportable single-user operating system (in Pascal) for the PDP-11, Intel 8080, and Ti-990. While at the Human Performance Center at U of M, designed and implemented a real-time control system for the IBM 1800, to run an experimental psychology laboratory.
Other Experience	SAPI (speech software) developer, Palm software developer, Web developer, CORBA ORB developer, signal processing, communications processing.

Consulting and Contracting Engagements Since 1992

1999-Present	Consultant with Johnson-Laird, Inc. Software forensics analyst.
1999-2000	Consultant with GemStone to assist their client, Proxima Ltd. in using GemStone's system in the design of Proxima's billing application.
1998-1999	Contractor to Interval Research Corporation to deliver a compiler, linker and assembler for a StrongARM processor.
1997	Contractor to GemStone to serve as system architect of a system under development by IBM Japan, to provide a physician's order entry system at a teaching hospital in Fukuoka, Japan.
1992-1996	Contractor to USAirways. Lead designer, with USAirways Operations Research staff to produce four business decision support systems.

Employment History

9/90 – 7/97	Co-founder, Reusable Solutions Inc. Portland Oregon
3/88 - 9/90	Co-founder and President Instantiations Inc. Portland Oregon
5/86 - 2/88	Object Services Project Leader, System Concepts Lab Xerox PARC/Northwest, Portland, Oregon
5/85 - 5/86	Director of Marketing GemStone Corp., Portland Oregon
3/82 - 5/85	System Architect GemStone Corp., Portland Oregon
8/81 - 2/82	Software Engineer, Design Automation Division Tektronix Inc., Portland Oregon
1/76 - 7/81	Computer Scientist Comshare Inc., Ann Arbor Michigan
7/75 - 12/75	Contract Programmer, Human Performance Center University of Michigan, Ann Arbor Michigan
9/74 - 6/75	Staff Programmer, System Development Data General Corp., Westboro Massachusetts
5/70 - 8/74	Programmer - Human Performance Center University of Michigan, Ann Arbor Michigan